PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-225694
(43)Date of publication of application : 22.08.1995

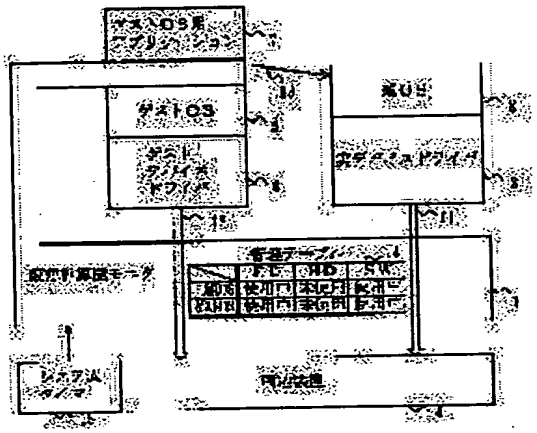
(51)Int.Cl. G06F 9/46

(21)Application number : 06-015089
(22)Date of filing : 09.02.1994

(71)Applicant : HITACHI LTD
(72)Inventor : YAMADA YOSHIHIRO
MAKIOKA JUNICHI

(54) VIRTUAL COMPUTER SYSTEM

(57)Abstract:
PURPOSE: To provide a virtual computer system capable of smoothly performing a processing even when the same peripheral equipment is alternately utilized from plural OSes.
CONSTITUTION: A virtual computer 1 is provided with a table 14 for storing the state of the peripheral equipment 4 controlled by a real device controller 3 and the state of the peripheral equipment controlled by a quest device driver 6. Then, by monitoring the respective device drivers 3 and 6 accessing the peripheral equipment 4 by the virtual machine monitor 1, checking the table 14 for storing the states of the peripheral equipment and permitting access only when conditions are valid, the access of the peripheral equipment by the plural OSes is smoothly performed.



LEGAL STATUS
[Date of request for examination]
[Date of sending the examiner's decision of r jection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's d cision of rejection]
[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号 ✓

特開平7－225694

(43)公開日　平成７年(1995)８月22日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　9/46	３５０	7737－5B		

審査請求　未請求　請求項の数３　ＯＬ　（全７頁）

(21)出願番号　特願平6－15089

(22)出願日　平成６年(1994)２月９日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(72)発明者　山田　善弘
神奈川県横浜市戸塚区吉田町292番地株式会社日立製作所映像メディア研究所内

(72)発明者　牧岡　淳一
茨城県日立市東多賀町一丁目１番１号株式会社日立製作所情報映像メディア事業部内

(74)代理人　弁理士　小川　勝男

(54)【発明の名称】　仮想計算機システム

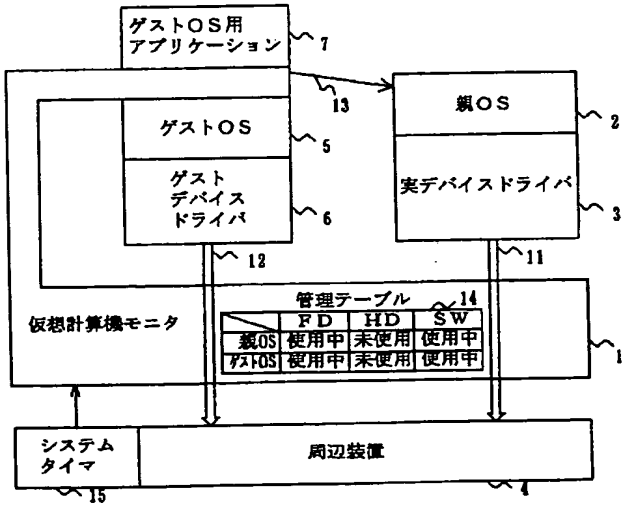
(57)【要約】

【目的】複数のＯＳから同一の周辺機器を交互に利用しても円滑に処理の行える仮想計算機システムを提供すること。

【構成】仮想計算機モニタに実デバイスコントローラにより制御される周辺装置の状態と、ゲストデバイスドライバにより制御される周辺機器の状態を格納するテーブルを設ける。

【効果】各デバイスドライバが周辺装置をアクセスするのを仮想計算機モニタで監視し、周辺機器の状態を格納するテーブルをチェックして、条件が成立する場合のみアクセスを許可することにより、複数のＯＳによる周辺機器のアクセスが円滑に行うことができる。

図１



1

【特許請求の範囲】

【請求項１】実計算機を制御する親ＯＳと、該親ＯＳの動作を監視する仮想計算機モニタと、該仮想計算機モニタの制御のもとで動作する一つ以上のゲストＯＳと、ゲストＯＳで動作するアプリケーションから成り、ゲストＯＳ上のアプリケーションから親ＯＳの機能を直接利用することのできるインターフェイスを有することを特徴とする仮想計算機システム。

【請求項２】実計算機を制御する親ＯＳと、該親ＯＳの動作を監視する仮想計算機モニタと、該仮想計算機モニタの制御のもとで動作する一つ以上のゲストＯＳと、親ＯＳで動作するアプリケーションから成り、親ＯＳ上のアプリケーションからゲストＯＳの機能を直接利用することのできるインターフェイスを有することを特徴とする仮想計算機システム。

【請求項３】実計算機を制御する親ＯＳと、周辺装置を制御する実デバイスハンドラと、該親ＯＳの動作を監視する仮想計算機モニタと、該仮想計算機モニタの制御のもとで動作する一つ以上のゲストＯＳと、ゲストＯＳから実計算機の周辺装置を制御するゲストデバイスドライバとゲストＯＳで動作するアプリケーションから成り、前記仮想計算機モニタに実デバイスコントローラにより制御される周辺装置の状態と、ゲストデバイスドライバにより制御される周辺機器の状態を格納するテーブルを設け、前記テーブルの内容を監視して実際の周辺機器を操作する手段を設けたことを特徴とする仮想計算機システム。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、一つの実計算機上で複数の異なるＯＳを利用する仮想計算機に係わり、特に、ゲストＯＳ上のアプリケーションから親ＯＳのサービスを直接利用する事ができ、さらに、ゲストＯＳと親ＯＳから効率良く実計算機の周辺機器を制御することのできる仮想計算機システムに関する。

【０００２】

【従来の技術】複数のＯＳを同時に動かすような仮想計算機システムにおいては、実計算機のハードウエアに対して、２つ以上のＯＳから同時に入出力要求が発生しないようにするのが普通である。例えば特開平３－１３１９３６号公報では、仮想計算機モニタに、ゲストＯＳによる入出力装置へのアクセスを管理するためのテーブルを設けて２つ以上のＯＳからの入出力装置への要求が同時に起きないように管理している。

【０００３】しかし、従来の方式では、ゲストＯＳ上のアプリケーションが、親ＯＳのサービスによる周辺装置へのアクセスと、ゲストＯＳのサービスによる周辺装置へのアクセスを交互に用いる必要がある場合には、周辺装置へのアクセスごとに各ＯＳが前記周辺装置を使いはじめてから処理を終了して解放するまでに別のＯＳによるアクセスができず、処理を円滑に行えない可能性が有った。

2

【０００４】

【発明が解決しようとする課題】本発明の目的は、複数のＯＳを同時に動かすことができ、ゲストＯＳ上のアプリケーションからゲストＯＳのサービスと親ＯＳのサービスを交互に用いて一つの周辺装置をアクセスした場合でも円滑に周辺機器の利用が可能な仮想計算機システムを提供することにある。

【０００５】

【課題を解決するための手段】前記課題を解決するために、本発明では、実計算機を制御する親ＯＳと、該親ＯＳの動作を監視する仮想計算機モニタと、該仮想計算機モニタの制御のもとで動作する一つ以上のゲストＯＳと、ゲストＯＳで動作するアプリケーションから成る仮想計算機システムにおいてゲストＯＳ上のアプリケーションから親ＯＳの機能を直接利用することのできるインターフェイスを設ける。また、実計算機を制御する親ＯＳと、該親ＯＳの動作を監視する仮想計算機モニタと、該仮想計算機モニタの制御のもとで動作する一つ以上のゲストＯＳと、親ＯＳで動作するアプリケーションから成る仮想計算機システムにおいて親ＯＳ上のアプリケーションからゲストＯＳの機能を直接利用することのできるインターフェイスを設ける。さらに、実計算機を制御する親ＯＳと、周辺装置を制御する実デバイスハンドラと、該親ＯＳの動作を監視する仮想計算機モニタと、該仮想計算機モニタの制御のもとで動作する一つ以上のゲストＯＳと、ゲストＯＳから実計算機の周辺装置を制御するゲストデバイスドライバとゲストＯＳで動作するアプリケーションから成る仮想計算機システムにおいて、前記仮想計算機モニタに実デバイスコントローラにより制御される周辺装置の状態と、ゲストデバイスドライブにより制御される周辺機器の状態を格納するテーブルを設け、前記テーブルの内容を監視して実際の周辺機器を操作する手段を設ける。

【０００６】

【作用】実計算機を制御する親ＯＳと、該親ＯＳの動作を監視する仮想計算機モニタと、該仮想計算機モニタの制御のもとで動作する一つ以上のゲストＯＳと、ゲストＯＳで動作するアプリケーションから成る仮想計算機システムにおいて、ゲストＯＳ上のアプリケーションから特定の割込を発生させることにより、仮想計算機モニタに制御を移すようにする。仮想計算機モニタは、特定の割込が発生すると、親ＯＳを起動して親ＯＳのサービスを要求してサービスを行わせる。親ＯＳのサービスが終了した後にゲストＯＳに復帰することにより、ゲストＯＳ上のアプリケーションから親ＯＳの機能を直接利用することができるようにする。

【０００７】また、実計算機を制御する親ＯＳと、該親ＯＳの動作を監視する仮想計算機モニタと、該仮想計算

機モニタの制御のもとで動作する一つ以上のゲストＯＳと、親ＯＳで動作するアプリケーションから成る仮想計算機システムにおいて、親ＯＳ上のアプリケーションから特定の割込を発生させることにより、仮想計算機モニタに制御を移すようにする。仮想計算機モニタは、特定の割込が発生すると、ゲストＯＳを起動してゲストＯＳのサービスを要求してサービスを行わせる。ゲストＯＳのサービスが終了した後に親ＯＳに復帰することにより、親ＯＳ上のアプリケーションからゲストＯＳの機能を直接利用することができるようにする。

【０００８】さらに、ゲストＯＳ上のアプリケーションから親ＯＳを通じての周辺機器のアクセスとゲストＯＳを通じての周辺機器のアクセスを交互に行う場合でも、仮想計算機モニタに実デバイスコントローラにより制御される周辺装置の状態と、ゲストデバイスドライバにより制御される周辺機器の状態を格納するテーブルを設け、各デバイスドライブが周辺装置をアクセスするのを仮想計算機モニタで監視して、どちらかのデバイスドライバが周辺機器にアクセスする時点で、周辺機器の状態を格納するテーブルをチェックして、条件が成立する場合のみ周辺機器に対するアクセスを許可することにより、複数のＯＳによる周辺機器のアクセスが円滑に行えるようにする。

【０００９】

【実施例】図１は、本発明の一実施例の構成図である。図中、構成要素１は仮想計算機モニタ、２は親ＯＳ、３は実デバイスドライバ、４は実計算機の周辺装置、５はゲストＯＳ、６はゲストデバイスドライバ、７はゲストＯＳ用アプリケーション、１１は実デバイスドライバからのハードウエアアクセス、１２はゲストデバイスドライバからのハードウエアへのアクセス、１３はゲストＯＳ用アプリケーションからの親ＯＳの呼出、１４は仮想計算機モニタ内の周辺機器の状態管理テーブル、１５はシステムタイマである。

【００１０】図２は、本発明の動作を示すフロー図である。図中、番号２００から２１５は、処理内容を示す。

【００１１】仮想計算機モニタ１は、親ＯＳ２およびゲストＯＳ５よりも高い特権を持っており、各ＯＳの動作を監視するものとする。立ちあげ時はまず、親ＯＳをはじめに立ちあげる。親ＯＳは特権レベルの高いスーパーバイザモードをサポートしないものとする。親ＯＳを立ちあげてから親ＯＳにより仮想計算機のプログラムを起動する。仮想計算機のプログラム自体がスーパーバイザモードを利用してしまうことにより仮想計算機モニタ１が親ＯＳよりも高い特権レベルを持つことができる。なお、ゲストＯＳもスーパーバイザモードを用いないと仮定して以後の説明を続ける。仮想計算機モニタ１はＣＰＵからのＩ／Ｏアクセスを監視することにより周辺装置へのアクセスを検出する。また、ソフトウエア割り込みを監視することによりアプリケーションからのＯＳに対する呼出を検出する。さらに、仮想計算機モニタはハードウエア割り込みを監視することにより周辺機器からの割込を検出する。

【００１２】親ＯＳ２は周辺装置４を有する実計算機を用いるために設計された本来のＯＳであり、実デバイスドライバ３というプログラムにより周辺装置４の制御を行う。矢印１１は実デバイスドライバによる周辺機器４へのアクセスを示す。アクセス１１は前述のように仮想計算機モニタにより監視されている。

【００１３】ゲストＯＳ５は、親ＯＳ２用のハードウエアとは異なるハードウエアを対象に作られたＯＳであり、ゲストＯＳ用アプリケーション７を動かすことができる。ゲストＯＳ５はゲストデバイスドライバ６により周辺装置４にアクセスする。ここでは、ゲストデバイスドライバ６は周辺装置４を制御可能なように作られているものとする。

【００１４】本来、ゲストＯＳ用アプリケーション７を周辺装置４を有する実計算機で動作させるためには、アプリケーション７を親ＯＳ用に作り替えることが必要である。しかし、アプリケーション７の規模が大きい場合は、アプリケーション７の変更のための手間が非常に大きなものになってしまう。そこで、周辺装置４を有する実計算機の上で仮想計算機モニタ１を動作させ、仮想計算機モニタ１の管理の下でゲストＯＳ５およびゲストＯＳ５用のアプリケーションを利用することにより、異なるハードウエアの上でもアプリケーション７を利用することを可能とする。

【００１５】計算機のハードウエアが異なると、メモリマップが異なる場合が多い。そこで仮想計算機モニタ１は、ゲストＯＳ５およびゲストＯＳ用アプリケーション７が動作するためのメモリマップを実現する機能を有するものとする。

【００１６】はじめに、ゲストＯＳ用アプリケーション７がゲストＯＳ５に対してサービスを要求した場合の動作から説明する。ゲストＯＳ用アプリケーション７はゲストＯＳ５に対してソフトウエア割り込みの形でＯＳのサービスを要求する。ＯＳのサービスとしては例えば外部記憶装置に対する読み書きの要求や、入出力装置に対する入出力の要求などが有る。

【００１７】処理２００においてソフトウエアの割込命令が実行されると、処理２０１において仮想計算機モニタ１が起動される。仮想計算機モニタ１は割込の種類を判断し、それがゲストＯＳ５に対するサービスの要求の場合は、処理２０２のゲストＯＳ５の割込処理ルーチンに制御を移す。ゲストＯＳ５は処理２０３のゲストデバイスドライバ６を利用して周辺装置４をアクセスし、サービスを実行する。サービス終了後に、処理２０５でゲストＯＳ５からゲストＯＳ用アプリケーション７に復帰する。

【００１８】次に、ゲストＯＳ用アプリケーション７か

ら親ＯＳ２のサービスを要求する場合の説明を行う。処理２００でゲストＯＳ用アプリケーション７はソフトウエア割り込みの形でＯＳのサービスを要求する。ソフトウエアの割込命令が実行されると、処理２０１で仮想計算機モニタ１が起動される。仮想計算機モニタ１は割込の種類を判断する。ここでは、特定の割込番号をあらかじめ決めておき、特定の割込番号の場合は親ＯＳ２のサービスを要求する割込とわかるようにしておく。割込が発生して、それが親ＯＳ２に対するサービスの要求の場合は、仮想計算機モニタ１は処理２１２で、親ＯＳ２の割込処理ルーチンを実行させるためにまず、親ＯＳ２の動作可能なメモリマップを設定してから親ＯＳ２の割込処理ルーチンに制御を移す。親ＯＳ２は処理２１３で実デバイスドライバ３を利用して周辺装置４をアクセスし、サービスを実行する。サービス終了後に、仮想計算機モニタ１を呼び出すプログラムを実行して仮想計算機モニタ１に制御を移す。ここでは、また、割込処理から復帰する命令を仮想計算機モニタ１で監視して割込処理の終了を検出しても良い。仮想計算機モニタ１はゲストＯＳ２が実行可能なメモリマップに設定してからゲストＯＳ用アプリケーション７に復帰する。

【００１９】以上の方式で、ゲストＯＳ用アプリケーション７から親ＯＳ２のサービスを受けることが可能となる。この方式の利点は、例えばゲストＯＳ５で用いるフロッピディスクのファイル構造と親ＯＳで用いるフロッピディスクのファイル構造が異なる場合でも、一つのディスクドライブにゲストＯＳ用のフロッピディスクをいれておきもう一つのフロッピディスクドライブに親ＯＳ用のフロッピディスクをいれておき、ゲストＯＳのアプリケーション７が自由に両方のデータ構造を有するフロッピディスクにアクセスしてデータやファイルの読み書きを行うことを可能とすることができる。

【００２０】次に、２個以上の異なるＯＳとデバイスドライブにより、一つの周辺機器にアクセスする場合の問題点およびその解決手段について説明する。一般的な方法としては、一つのＯＳが周辺機器をアクセスする場合は、別のＯＳはその周辺機器に対するアクセスは一切禁止してしまうのが普通であり、これを排他制御と呼ぶ。ところが、例えばフロッピディスクやハードディスクのように媒体をモータで回転させてデータを読み書きする周辺機器では、データの読み書きが終了しても数秒から数分の間モータを回し続けることが多い。これはモータの起動および停止には有る程度の時間を必要とするので、一度フロッピディスクへのアクセスが行われてモータを回転させた場合は、しばらくの間回転させた状態にし、フロッピディスクへのアクセスが一定期間の間になかった場合のみ消費電力の低減やディスクドライブの機械的な摩耗を防ぐためにモータを止めるようにしているシステムが多い。このモータの起動および停止の制御をデバイスドライバによりソフトウエアで行っている場合

は問題が起きる可能性が有る。

【００２１】本実施例における実計算機システムにおいては、フロッピディスクのモータの制御は、システムタイマー１５により行われる。デバイスドライバ３により、フロッピディスクの読み書きを行う場合、デバイスドライバ３はまず、デバイスドライバ３のワークエリア内のカウンタを特定の値に設定してからフロッピディスクのモータを起動状態にする。モータの起動にはある程度の時間がかかるので時間待ちを行ってからフロッピディスクの読み書きを行う。

【００２２】次にフロッピディスクの読み書きを行う場合には、モータの状態を検出して、モータがオフ状態の場合はカウンタを設定してからモータを起動して読み書きを行う。また、モータが回転している状態の場合はカウンタを一定の値に設定してすぐに読み書きを行うことができる。

【００２３】モータの停止はシステムタイマー割り込みにより行う。実計算機の周辺装置４にはシステムタイマ１５が設けられており、１秒間に数十回の割合でシステムタイマ割り込みを発生する。実計算機の場合は、システムタイマ割込が発生すると、親ＯＳのシステムタイマ割込処理ルーチンに制御が移されシステムタイマの処理が行われる。システムタイマ割込でフロッピディスクのモータ制御を行う場合は、システムタイマ割り込み処理ルーチンでまず、モータ制御用のカウンタの値を１だけ減らす。もしも、０になった場合は、モータを停止するようにする。ここで、システムタイマが１秒に２０回発生する場合、フロッピディスクのモータを１０秒間アクセスが無かった場合に停止したい場合は、デバイスドライバによりカウンタの初期値を２００に設定すれば良い。

【００２４】ゲストＯＳ５によりフロッピディスクをアクセスする場合も親ＯＳとほぼ同じシーケンスで制御を行う。

【００２５】ここで、一つのディスクドライブに、親ＯＳ２のファイル構造のフロッピディスクをセットして、もう一つのディスクドライブにゲストＯＳのファイル構造を有するフロッピディスクをセットし、ゲストＯＳ用のアプリケーション７から交互にゲストＯＳ５と親ＯＳ２のサービスを利用して両方のフロッピディスクにアクセスする場合を考える。ここで、ゲストデバイスドライバ６と実デバイスドライバ３はそれぞれの内部にモータ制御用のカウンタを用意しており、システムタイマ１５からの割込がかかると、仮想計算機モニタによりゲストＯＳ５側の割込処理ルーチンおよび親ＯＳ２側の割込処理ルーチンの両方の割込処理ルーチンによる処理を行うものとする。また、ディスクドライブのモータはどのドライブも同時に起動および停止の制御を受けるものとする。

【００２６】ゲストＯＳ用のアプリケーション７が親Ｏ

Ｓ２により親ＯＳ２のファイル構造を有するフロッピディスクの読み書きを行った後で、ゲストＯＳ５により別のディスクドライブにセットされたゲストＯＳ５のファイル構造を有するフロッピディスクをアクセスしている場合、親ＯＳ２によりフロッピディスクをアクセスしてからしばらくすると、システムタイマ１５による割込処理ルーチンにより、ゲストＯＳ５でフロッピディスク使用中にもかかわらず強制的にモータを停止してしまう可能性がある。

【００２７】前述の問題は、仮想計算機モニタに両方のＯＳによるモータの状態を保持するテーブルを設けて、どちらかのＯＳのデバイスドライバがモータを停止しようとした場合に、前記のテーブルの内容をチェックして、どのＯＳのデバイスドライバにおいてもモータが停止になる場合のみ実際にモータの停止を行うようにすることで解決することができる。

【００２８】仮想計算機モニタ１は内部に管理テーブル１４を持ち、さらに、実デバイスドライバ３からの周辺装置４へのアクセス１１およびゲストデバイスドライバ６からの周辺装置４へのアクセス１２を監視する。これは、ＣＰＵが特定のＩ／Ｏポートへのアクセスを検出する機能を利用することにより実現する。

【００２９】フロッピディスクのモータを制御するＩ／Ｏポートへのアクセスが有った場合に、仮想計算機モニタ１呼び出されるようにしておき、モータ起動時は、管理テーブル１４にどのＯＳでモータを起動状態にしたかという情報を設定する。どのＯＳからＩ／Ｏポートがアクセスされたのかを知るためには、現在動作中のＯＳを識別するためのフラグを設けて、仮想８６モニタによりそのフラグを管理する方法がある。また、Ｉ／Ｏポートアクセス時に実行されていたプログラムの置かれているアドレスを調べることにより、どのＯＳが実行中かを識別することも可能である。

【００３０】システムタイマ１５によるタイマ割込により、仮想計算機モニタ１はゲストＯＳ５の割込処理ルーチンと親ＯＳ２の割込処理ルーチンの両方の処理を行わせる。モータの停止は、システムタイマ１５がタイマ割込を発生することにより割込処理ルーチンがカウンタをへらし、カウンタの値が０になった場合に行われる。割込処理ルーチンがＩ／Ｏポートをアクセスしてモータを停止しようとした場合に仮想計算機モニタはそれを検出して管理テーブル１４に、どのＯＳでモータが停止状態になるかを設定する。次に、他のＯＳによるモータの状況を調べて、全てのＯＳにおいてモータの状態が停止状態になることを確認した場合のみ実際にＩ／Ｏポートをアクセスしてモータを停止状態にする。なお、他のＯＳで一つでもモータが回転状態になっている場合は、テーブル１４を設定するだけで実際にはＩ／Ｏポートへのアクセスは行わないようにする。

【００３１】本発明は外部記憶装置以外にも、リレーにより計算機自身が計算機の電源を遮断するようなシステムにおいても応用できる。例えば親ＯＳ２が動作中にゲストＯＳ５から電源が遮断されるとシステムに障害が発生する可能性があるが、仮想計算機モニタ１において、各ＯＳの電源遮断状況を持つフラグを用意して、全てのＯＳにおいて電源が遮断されたときにのみ実計算機の電源を遮断することにより障害の発生を回避することが可能となる。

【００３２】図３は、本発明の一実施例の構成図である。図中、構成要素１～１５は図１の構成要素と同様の要素を示すので説明は省略する。３０７は親ＯＳ用アプリケーション、３１３は親ＯＳ用アプリケーションからのゲストＯＳの呼出である。

【００３３】通常、親ＯＳのアプリケーションがゲストＯＳのサービスを必要とすることはあまり無いが、親ＯＳの開発環境がゲストＯＳのそれよりも優れている場合は、親ＯＳ上でゲストＯＳのアプリケーションのサポート用ソフトウエアを開発することが有り得る。例えば、ゲストＯＳ用のフロッピディスクの読み書きが必要な場合、親ＯＳ上でうごくアプリケーションからゲストＯＳのサービスを利用したい場合が出てくる。

【００３４】その場合、親ＯＳのアプリケーションから、ゲストＯＳのサービスを要求する手段が必要となる。次に、親ＯＳ用アプリケーション３０７がゲストＯＳ５のサービスを要求する場合の説明を行う。親ＯＳ用アプリケーション３０７はソフトウエア割り込みの形でＯＳのサービスを要求する。ソフトウエアの割込命令が実行されると、仮想計算機モニタ１が起動される。仮想計算機モニタ１は割込の種類を判断する。ここでは、特定の割込番号をあらかじめ決めておき、特定の割込番号の場合はゲストＯＳ５のサービスを要求する割込とわかるようにしておく。割込が発生して、それがゲストＯＳ５に対するサービスの要求の場合は、仮想計算機モニタ１は、ゲストＯＳ５の割込処理ルーチンを実行させるためにまず、ゲストＯＳ５の動作可能なメモリマップを設定してからゲストＯＳ５の割込処理ルーチンに制御を移す。ゲストＯＳ５はゲストデバイスドライバ６を利用して周辺装置４をアクセスし、サービスを実行する。サービス終了後に、仮想計算機モニタ１を呼び出すプログラムを実行して仮想計算機モニタ１に制御を移す。ここでは、また、割込処理から復帰する命令を仮想計算機モニタ１で監視して割込処理の終了を検出しても良い。仮想計算機モニタ１は親ＯＳ２が実行可能なメモリマップに設定してから親ＯＳ用アプリケーション３０７に復帰する。

【００３５】以上の方式で、親ＯＳ用アプリケーション３０７からゲストＯＳ５のサービスを受けることが可能となる。この方式の利点は、例えば親ＯＳ２で用いるフロッピディスクのファイル構造とゲストＯＳ５で用いるフロッピディスクのファイル構造が異なる場合でも、一

つのディスクドライブにゲストＯＳ用のフロッピディスクをいれておきもう一つのフロッピディスクドライブに親ＯＳ用のフロッピディスクをいれておき、親ＯＳのアプリケーション２が自由に両方のデータ構造を有するフロッピディスクにアクセスしてデータやファイルの読み書きを行うことを可能とすることなどに有る。

【００３６】

【発明の効果】本発明によれば、実計算機を制御する親ＯＳと、該親ＯＳの動作を監視する仮想計算機モニタと、該仮想計算機モニタの制御のもとで動作する一つ以上のゲストＯＳと、ゲストＯＳで動作するアプリケーションから成る仮想計算機システムにおいて、ゲストＯＳ上のアプリケーションから特定の割込を発生させることにより、仮想計算機モニタに制御を移すようにし、親ＯＳを起動して親ＯＳのサービスを行わせる。親ＯＳのサービスが終了した後にゲストＯＳに復帰することにより、ゲストＯＳ上のアプリケーションから親ＯＳの機能を直接利用することができるようにする。

【００３７】また、実計算機を制御する親ＯＳと、該親ＯＳの動作を監視する仮想計算機モニタと、該仮想計算機モニタの制御のもとで動作する一つ以上のゲストＯＳと、親ＯＳで動作するアプリケーションから成る仮想計算機システムにおいて、親ＯＳ上のアプリケーションから特定の割込を発生させることにより、仮想計算機モニタに制御を移すようにし、ゲストＯＳを起動してゲストＯＳのサービスを行わせる。ゲストＯＳのサービスが終了した後に親ＯＳに復帰することにより、親ＯＳ上のアプリケーションからゲストＯＳの機能を直接利用することができるようにする。

【００３８】さらに、ゲストＯＳ上のアプリケーションから親ＯＳを通じての周辺機器のアクセスとゲストＯＳを通じての周辺機器のアクセスを交互に行う場合でも、仮想計算機モニタに実デバイスコントローラにより制御される周辺装置の状態と、ゲストデバイスドライバにより制御される周辺機器の状態を格納するテーブルを設け、各デバイスドライバが周辺装置をアクセスするのを仮想計算機モニタで監視して、どちらかのデバイスドライバが周辺機器にアクセスする時点で、周辺機器の状態を格納するテーブルをチェックして、条件が成立する場合のみ周辺機器に対するアクセスを許可することにより、複数のＯＳによる周辺機器のアクセスが円滑に行うことができる。

【図面の簡単な説明】

【図１】本発明の１実施例を示す構成図である。

【図２】処理内容の説明図である。

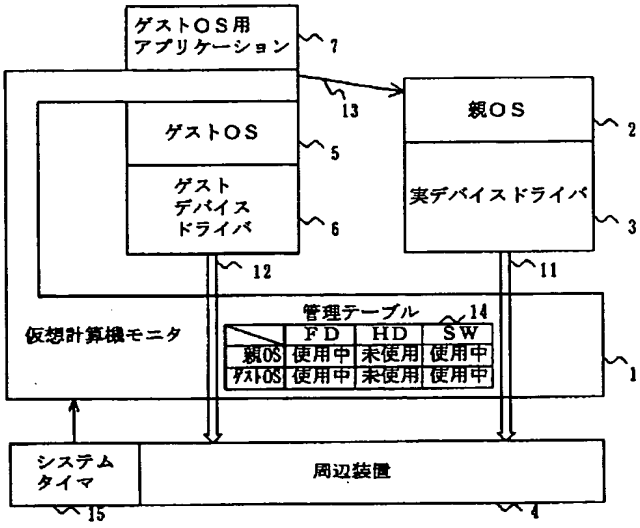
【図３】本発明の１実施例を示す構成図である。

【符号の説明】

１…仮想計算機モニタ、
２…親ＯＳ、
３…実デバイスドライバ、
４…周辺装置、
５…ゲストＯＳ、
６…ゲストデバイスドライバ、
７…ゲストＯＳ用アプリケーション、
１１…実デバイスドライバによるアクセス、
１２…ゲストデバイスドライバによるアクセス、
１３…親ＯＳへのインターフェイス、
１４…管理テーブル、
２００～２１５…処理内容、
３０７…親ＯＳ用アプリケーション、
３１３…ゲストＯＳへのインターフェイス。

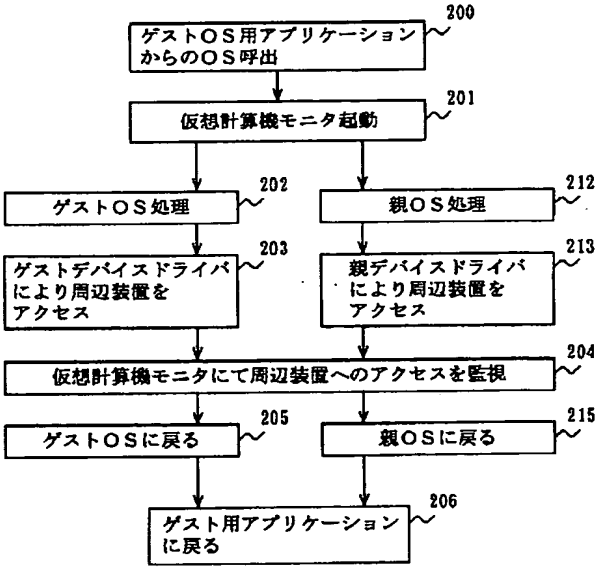
【図１】

図１



【図２】

図２



【図３】

図３